

Proposed Approximate Dynamic Programming for Pathfinding under Visible Uncertainty

Bryan A. Knowles and Mustafa Atici

December 3, 2015

Abstract

Continuing our preliminary work [2], we define the safest-with-sight pathfinding problems and explore its solution using techniques borrowed from measure-theoretic probability theory. We find a simple recursive definition for the probability that an ideal pathfinder will select an edge in a given scenario of an uncertain network where edges have probabilities of failure and vertices provide “vision” of edges via lines-of-sight. We propose an approximate solution based on our theoretical findings that would borrow techniques from approximate dynamic programming.

1 Introduction

We introduce a probabilistic-decision variant of the classic pathfinding problem defined on a directed acyclic graph [8] where each edge has some probability of “failure” and each vertex has “vision” of a set of edges. That is, once the pathfinder has reached a vertex, it can “know” whether the edges within the sight of that vertex are up or down; the status of these edges are said to not change during the duration of a single “trial.” How, then, should the pathfinder behave if it wishes to take the “safest” (most likely to succeed) path taking this “sight” into account?

Although this safest-with-sight problem, as we will refer to it throughout, is simple to state, and we have restricted it to directed acyclic graphs, which generally reduce the complexity of problems, we believe that the introduction of vision into the mix makes this problem computationally hard.

Naturally, this being a graph-theory-grounded problem, we wish to determine a greedy algorithm [8] to answer the query, “given a current scenario, will an *ideal* pathfinder’s *one next* move be x ?” However, this being a problem steeped in uncertainties, we must also accomplish this task probabilistically, defining a decision function by borrowing techniques from measure-theoretic probability theory [5]. So, to calculate the solution in the general case, where we find that a greedy solution cannot work, we find a dynamic algorithm and propose, to reduce the computational complexities in that unmodified algorithm, to use an approximate dynamic scheme instead [6].

In section 2 we briefly discuss a portion of the literature on the subject, in section 3 we discuss our laying the groundwork for a theoretical solution to this problem, and in section 4 we conclude by stating our goals in implementing an approximate solution for the general case that is exact and polynomial in certain cases of the problem.

2 Previous Work

We have been unable to find any publication on the same problem as ours or on a problem that is immediately reducible to safest-with-sight. The key difference between our definition and others is that of vision—the ability of the pathfinder to select options that will make future options more informed. It is by this distinction that we mean “visible uncertainty.”

However, the literature does contain works related to *components* of ours. In a preliminary article [2], we explore several works on undirected, random, and directed-acyclic graphs each with edge-risk probabilities, and others on paths through and relationships between layers of probabilistic networks.

Having quickly reached problems with approaching safest-with-sight from a purely graph-theoretical perspective after releasing our preliminary work in preprint, we looked further into uncertainty and probability theory itself.

In a now classic work, Bart Kosko introduces fuzzy logic, now a staple in the machine learning literature, which does away with the binary of true and false in favor of a model based on partial membership within sets [3]. David Pollard writes on measure-theoretic probability theory, which reevaluates how we construct probabilities as not densities, but as measure functions or expected values of inclusion within a set of outcomes [5]. And in a collection of works by Springer, the foundations of applying fuzzy logic and uncertainty is set forth [7].

None of these publications has been more influential on our research more than Pollard’s, as it forced us to work axiomatically from set theoretical definitions, making clear the exact influence that vision has on “the math.” It is in the following statement, and no more, that vision has its principle effects: I will never (probability equals zero) decide to take an edge that is both down and in my accumulated line-of-sight.

3 Theoretical Foundations

Problem. An instance of the *safest-with-sight* problem is defined by the tuple (G, β, t_{sd}) . First, $G = (V, E, W)$, where V is a set of vertices, E is a set of edges and $ij \in E$ implies an edge exists between i and j such that $i < j$, and W is a set of lines-of-sight and $ijk \in W$ implies vertex i has line-of-sight to edge jk such that $i \leq j$. Second, β is a set of parameters where $\beta_{\alpha ij}$ is the probability of event α_{ij} , in which edge ij is down or obstructed. Finally, we are given the task t_{sd} , representing the starting vertex s and destination vertex d . Given a possible first step si that the pathfinder could take, assuming the pathfinder behaves ideally for the following properties, decide whether the pathfinder will take that first step:

1. if the pathfinder crosses an edge that is down, it can no longer move and fails the trial immediately
2. the pathfinder can only follow directed, simple paths that start at the starting point and end either at the destination or the first visited point at which the pathfinder knows that all remaining paths end in “dead-ends”
3. the pathfinder will never attempt to cross edges that it knows are down or lead only to dead-ends
4. the pathfinder always selects the path that maximizes its probability of successfully reaching the destination with respects to its current knowledge
5. the pathfinder always knows the layout of the graph, including lines-of-sight
6. the pathfinder only knows the up/down status of edges that are in a line-of-sight of any vertex it has ever visited during the current trial
7. extraneous edges have been removed, so any unobstructed path the pathfinder takes will lead it to the destination
8. the pathfinder stops at each vertex to consider new information from new lines-of-sight, and it may use this information to reroute its current course such that the probability of success remains optimized with respects to the pathfinder’s current knowledge
9. depending on the application, a definition of “tie-broken” is given that imposes a total ordering on the edges without respect to probabilities of success; if no definition for tiebreaking can be given, we suggest using the outward indices of edges for tiebreaking so that the pathfinder will select the optimal edge with the highest outward index

Theorem 1. The safest-with-sight problem cannot be solved with a greedy algorithm.

Proof. The probability of success of a “first edge” is determined, in part, by the probability of future edges and their probabilities of being taken; however, the probabilities of a future edge is likewise determined by the probabilities of first edges, since they determine what set of lines-of-sight may be available once that future edge has been reached. In other words, the behavior of the pathfinder at the first step and future steps are caught in a chicken-and-egg problem. This prevents safest-with-sight from meeting the greedy-choice property, since solutions to problems depend on solutions to subproblems [1]. Therefore, safest-with-sight cannot be greedy. *q.e.d.*

Definition. We define a function Φ that digitizes statements as 1s or 0s.

$$\Phi(x) = \begin{cases} 1 & x \text{ is true} \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

$$\text{if } x_i \text{ are independant then } \forall_i \Phi(x_i) = \prod_i \Phi(x_i) \quad (2)$$

Definition. We define P as an underlying a probability measure for solving this problem, choosing notation such that P maps *queries encoded as a summed series of terms* to the space \mathbf{B} of basis B . In-depth measure-theoretic details do not matter in our application, as we will replace all occurrences of P in our problem with either event methods, which correspond with computable functions, or with conditional statements of constant values.

First, our outcome space is defined as follows, letting each ω representing a vector or sum of terms where t is a term representing the task of the trial, each a is a term whether some edge was up or down, each d is a term describing whether some edge was taken or not taken by the pathfinder, and each g is a term describing the graph's edges, vertices, and lines-of-sight.

$$\Omega = \left\{ \omega \mid \omega = t + \sum a + \sum d + \sum g \right\} \quad (3)$$

Next, P , \mathbf{F} , and \mathbf{B} are defined as follows.

$$P : \mathbf{F} \rightarrow \mathbf{B} \quad (4)$$

$$\Omega \in \mathbf{F} \quad (5)$$

$$\text{if } \omega \in \Omega \text{ then } \{\omega\} \in \mathbf{F} \quad (6)$$

$$\text{if } A \in \mathbf{F} \text{ then } \Omega \setminus A \in \mathbf{F} \quad (7)$$

$$\text{if } A_1, A_2, \dots \in \mathbf{F} \text{ then } \bigcap_k A_k \in \mathbf{F} \quad (8)$$

$$\beta \cup \{0\} \subseteq \mathbf{B} \quad (9)$$

$$\text{if } b \in \mathbf{B} \text{ then } 1 - b \in \mathbf{B} \quad (10)$$

$$\text{if } b_1, b_2, \dots \in \mathbf{B} \text{ then } \prod_k b_k \in \mathbf{B} \quad (11)$$

And we state the following about P as it is used to encode queries about the pathfinder's behavior:

$$P(0) = 1 \quad (12)$$

$$P(a_1 + a_2 + \dots) = P(a_1 \text{ and } a_2 \text{ and } \dots) \quad (13)$$

$$P(\bar{a}) = 1 - P(a) \quad (14)$$

$$P(a + \bar{a}) = 0 \quad (15)$$

$$\text{if } a_i \text{ are independant then } \exists_i P(a_i) = 1 - \prod_i [1 - P(a_i)] \quad (16)$$

$$\exists_i [P(a_i + \xi_1 \mid b_i + \xi_2)] = \frac{\sum_i \left[P(a_i + \sum_{j=1}^{i-1} \bar{a}_j + b_i + \sum_{j=1}^{i-1} \bar{b}_j + \xi_1 + \xi_2) \right]}{\sum_i \left[b_i + \sum_{j=1}^{i-1} \bar{b}_j + \xi_2 \right]} \quad (17)$$

$$P(a) = \beta_a, \forall_{a \in \text{parametric terms}} \quad (18)$$

$$P(w + \dots) = 0, \forall_{w \in \text{contradictions}} \quad (19)$$

$$(20)$$

$$\text{if } i < s \text{ or } j > d \text{ then } P(t_{sd} + a_{ij} + \dots) = P(t_{sd} + \dots) \quad (21)$$

$$\text{ev}(a) = \sum_{a'} P(a' \mid a) P(\text{ev} \mid a') \quad (22)$$

$$\text{iff } a \text{ is M.R. on ev then } \text{ev}(a) = P(\text{ev} \mid a) \quad (23)$$

By the term M.R. above, we mean “maximally restrictive.” That is, the event method **ev**, given a , has no change in value for any b disjoint from a in $\text{ev}(a + b)$. Therefore, a contains as much information as possible for determining **ev**’s value.

Definition. We define the set of terms used to encode events as: t_{sd} , the task term; α_{ij} , terms for the event that edge ij is down; s_{ijk} , terms for the event that vertex i has vision of edge jk ; and δ_{ij} , terms for the event that the pathfinder *chose* to travel along edge ij , but not necessarily traveled along it safely.

Definition. We complete the definition of P by defining what we mean above by **parametric terms** and **contradictions**:

- the parametric terms are those for which we are given a β -parameter, such as α_{ij} in $\beta_{\alpha_{ij}}$
- if a and b are parametric terms, then $P(a \mid b) = P(a)$ unless $b = \bar{a}$, in which case $P(a \mid b) = 0$
- the contradictions are the minimum set of expression which, for all queries containing one or more of those expressions, the probability must be zero

This set of contradictions corresponds to exactly the following, derived directly from the properties of the problem and definition of P :

- classic–appealing to (15), $P(a + \bar{a}) = 0$
- restriction–appealing to property 1, $P(\delta_{ij} + \delta_{jk} + \alpha_{ij}) = 0$
- simplicity–appealing to property 2, $P(\delta_{ij} + \delta_{ik}) = 0$ and $P(t_{sd} + \delta_{jk} \sum_{ij} \bar{\delta}_{ij}) = 0$ if $j \neq s$
- refusal–appealing to property 3, $P(\delta_{ij} + \delta_{km} + \alpha_{km} + s_{ikm}) = 0$
- dead-ends–appealing to properties 2 and 3, $P(\delta_{ij} \mid t_{sd} + \dots) = 0$ if $j \neq d$ and $\forall_{jk} [P(\delta_{jk} \mid t_{sd} + \dots) = 0]$
- suboptimal–appealing to property 4, $P(\delta_{si} \mid t_{sd} + \dots) = \text{optimal}(\delta_{si} + t_{sd} + \dots) \times \text{tiebroken}(\delta_{si} + t_{sd} + \dots)$

Definition. We define a set of event methods **select** _{si} for all edges si that, on maximally restrictive input, equals 1 only where the pathfinder would select edge si given the current task t_{sd} , knowledge ξ , and lines-of-sight S ; otherwise, it equals 0. It is important to note that this event method is simply the functional equivalent of δ_{si} , only given an easier to express name. This definition requires the definitions of event methods **optimal** _{si} , **tiebroken** _{si} , and **success** _{si} , as well as S' , which corresponds to the lines-of-sight of a subproblem where the vision provided by S has been “copied” to all other vertices and truncated such that no vertex “sees behind itself.”

$$\text{select}_{si}(t_{sd} + S + \xi) = \text{optimal}_{si}(t_{sd} + S + \xi) \times \text{tiebroken}_{si}(t_{sd} + S + \xi)$$

$$\text{where } \text{optimal}_{si}(t_{sd} + S + \xi) = \forall_{sj} \Phi [\text{success}_{si}(t_{sd} + S + \xi) \geq \text{success}_{sj}(t_{sd} + S + \xi)]$$

$$\text{and } \text{success}_{sd}(t_{sd} + S + \xi) = P(\bar{\alpha}_{sd} \mid \delta_{sd} + t_{sd} + S + \xi)$$

$$\text{and } \text{success}_{si}(t_{sd} + S + \xi) = P(\bar{\alpha}_{si} \mid \delta_{si} + t_{sd} + S + \xi) \times \sum_{\xi'} \left[P(\xi' \mid \xi) \times \max_{ik} \text{success}_{ik}(t_{id} + S' + \xi') \right]$$

Lemma 1. If **success** _{si} is correctly the probability that the pathfinder will reach the destination (given only what it could know at s and assuming it has decided to traverse si) then **select** _{si} correctly determines whether the pathfinder will choose to traverse si .

Proof. Appealing to properties 3 and 9, the pathfinder will select an edge iff it is optimal and tiebroken. Obviously too, an edge is optimal iff there exists no other available edge that, according to the pathfinder’s current knowledge, has a higher probability of success.

We've defined optimal_{si} as a digitization of this notion, so when the input is maximally restrictive, this event method maps to $\{0, 1\}$. If we assume the input of select_{si} is maximally restrictive on both select_{si} and optimal_{si} , then, since the same input is passed to both, select_{si} will also map to $\{0, 1\}$, being itself the product of two such mappings. We may assume that the initial input, given by the problem, is maximally restrictive in this way, since it corresponds to precisely the knowledge held by the pathfinder while at s , the start of the trial. Because no other input is given to select_{si} , these assumptions about maximally restrictive input hold for select_{si} and optimal_{si} always. That is, there is no need to worry about weighted sums, as defined in (21) and (22).

Therefore, if success_{si} is correct, then optimal_{si} will yield the proper value and, in turn, select_{si} as well. *q.e.d.*

Lemma 2. Assuming the pathfinder has decided to traverse edge si , its probability of success (according only to the pathfinder's knowledge) is the probability it successfully crosses si and it succeeds from some subproblem where its starting point is instead i and lines-of-sight everywhere have been modified to include the lines-of-sight provided by s in the original problem.

Proof. When the pathfinder is traversing an edge, it can fail immediately. In the event that it does not, it, appealing to property 8, stops to consider new information. If s provided no lines-of-sight that were not provided by i , then its probability of success after crossing si is no different than its overall probability if it were to begin from i in the first place. If s *did* provide lines-of-sight useful after i that were not provided by i originally, because, appealing to property 6, the state of the network does not change during a single trial and the pathfinder does not forget information, we can model its "remembering" by copying the vision provided by s to all vertices ahead of it, adding no information to the subproblem that would not have already been known anyway. Therefore, by appealing to the properties of the problem, we can derive this lemma directly. *q.e.d.*

Theorem 2. success_{si} is correct as required by lemma 1.

Proof. First, consider what maximally restrictive input to success_{si} might look like: the maximum information that the pathfinder could have that could change the probability of its success, according only to the knowledge that it could have obtained via lines-of-sight, is where ξ contains α_{ij} or $\bar{\alpha}_{ij}$ terms for all edges ij referenced in S . This is precisely the sort of input assumed in lemma 1 to be passed to success_{si} by optimal_{si} .

Next, appealing to lemma 2 and (21), we write success_{si} as the following:

$$\text{success}_{si}(t_{sd} + S + \xi) = \sum_{\xi'} [P(\xi' \mid \xi) \times P(\bar{\alpha}_{si} \mid \delta_{si} + t_{sd} + S + \xi')] \times \quad (24)$$

$$\sum_{\xi'} \sum_{ik} [P(\xi' \mid \xi) \times \text{select}_{ik}(t_{id} + S' + \xi') \times \text{success}_{ik}(t_{id} + S' + \xi')] \quad (25)$$

Consider how the probability that an edge is up is only affected by whether the edge is already known to be up/down (referenced in S and α or $\bar{\alpha}$ is in ξ). Hence, the only properties of the problem that have an effect on $P(\bar{\alpha}_{si} \mid \dots)$ are (15) and the refusal contradiction; these require only terms that would already be in a maximally restrictive input to success_{si} . Therefore, for that the first weighted sum we can appeal to (23), at least only when the input to success_{si} is given to be maximally restrictive.

success_{si} is only ever given maximally restrictive input in our definitions, since optimal_{si} passes it input that is optimally restrictive and when success_{si} invokes success_{ik} , it does so only under a weighted sum iterating over all maximally restrictive inputs to the subproblem that contain ξ . Therefore, we can appeal to (23) for all cases covered by our definitions and rewrite the first weighted sum as just $P(\bar{\alpha}_{si} \mid \dots)$.

For the second weighted sum, note that ξ is not guaranteed to be maximally restrictive for the subproblem, since S' might reference an edge km such that neither α_{km} nor $\bar{\alpha}_{km}$ are in ξ ; therefore, the iteration over ξ' must be performed and cannot be simplified away as in the first weighted sum.

However, consider how the inner sum operates: because select_{si} maps to $\{0, 1\}$, and to 1 only where optimal and tiebroken, we can write this inner sum simpler as the maximum subproblem, where the notion of maximizing is with respects to both optimal_{ik} and tiebroken_{ik} .

Taking all of these simplifications into account, we produce exactly the recursive definition of success_{si} from above. With this in place, the base case definition is trivial to show: once the pathfinder has decided to

traverse some final edge leading directly from s into d , its probability of success is simply that of whether or not it successfully that edge—there is no “succeeds later” to worry about. Therefore, the recursive definition is just $P(\bar{\alpha}_{si} \mid \dots)$, what remains after simplifying the first weighted sum and removing the (in the base case) unnecessary second weighted sum.

Finally, since the first invocation of `successsi` is given the proper inputs, it is correct under lemma 2, it passes the correct inputs to future invocations, and the base case is also correct under lemma 2, `successsi` must be correct in determining the probability of success as required by lemma 1. *q.e.d.*

4 Conclusions and Future Work

The solution we have defined to the safest-with-sight problem, a decision problem based on `successsi`, is not computationally efficient. When each path through to a certain node would produce a different set of known edges, via accumulated lines-of-sight, then an exact algorithm would need to calculate each of those paths, a PCOUNT-time solution. In cases where the set of known edges at a node is always the same, then solutions to recursive cases of `successsi` could be cached using dynamic programming techniques and the runtime reduced to $O(E)$, since each edge would only need to be visited once by the algorithm, as in the following two examples:

- no lines-of-sight exist, so vision is always empty
- lines-of-sight are only of immediate neighbors, so vision cannot accumulate

With the cache-based improvement in mind, we propose an approximate solution to the general case of safest-with-sight, where solutions to recursive cases of `score` are cached and, when a *similar-enough* case has already been cached for the same edge, that cached solution would be used in hopes that the scores will not differ greatly. This approach will have to be done with care to find a balance between space required to cache results and time required to produce accurate solutions [6].

5 Special Thanks

Special thanks to IPython/Jupyter [4] for providing a free and open source platform used extensively when developing this work.

References

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Ronald Rivest. *Introduction To Algorithms*. Mcgraw-Hill College, 1990.
- [2] B. Knowles and M. Atici. Fault-Tolerant, but Paradoxical Path-Finding in Physical and Conceptual Systems. *ArXiv e-prints*, June 2014.
- [3] Bart Kosko. *Fuzzy Thinking*. Hyperion, 1993.
- [4] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.
- [5] David Pollard. *A User’s Guide to Measure Theoretic Probability (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, 2001.
- [6] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition (Wiley Series in Probability and Statistics)*. Wiley, 2011.
- [7] Springer. *Foundations of Reasoning under Uncertainty (Studies in Fuzziness and Soft Computing)*. Springer, 2010.
- [8] Robin J. Wilson. *Applications of Graph Theory*. Academic Press, 1980.